

Circular relationships and limiting allowed values

Source:

<http://www.tech-archive.net/Archive/Access/microsoft.public.access.tablesdbdesign/2005-12/msg00131.html>

- *From:* "Rararachel" <Rararachel@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Thu, 8 Dec 2005 13:27:03 -0800
-

Hi All,

I have a surprisingly simple, but complicated database requirement in that the database requires only 5 tables, but includes a circular reference by necessity. The problem I'm having is that I am unsure how to create the queries on which to base my input forms so that the correct data is displayed.

Essentially I have a workflow process that is described in the following way: A dataset is released which is deployed to several regions, and each deployment involves several tasks. Not every region receives every dataset, but every dataset is built and deployed to at least one region.

Several physical regions represented in the Region table
fldRegionID – pk
fldRegionName

Several sets of data represented in the Dataset table
fldDataID – pk
fldDataName
fldCycle (this represents the cycle with which the dataset is updated)

Each dataset has several tasks that need to be completed before they are deployed, some of which are also dependant on the region to which they are deployed. This is represented by two tables, one (build) which contains those tasks that are completed for all updates to the dataset, and the second (deployment) which contains those tasks that are required for each regions deployment. These are represented as follows:

– Build table
fldVersionID – pk
fldDatasetID pk + fk
fldReleaseDate
fldBuildTask1
fldBuildTask2
.
.
.

Circular relationships and limiting allowed values

fldBuildTaskn

– Deployment table

fldVersionID pk + fk

fldDatasetID pk + fk

fldRegionID pk + fk

fldDeployTask1

fldDeployTask2

.

.

.

fldDeployTaskn

Because each region may or may not receive an update, and the relationship between datasets and regions is m–m, a fifth, RegionDataset table is required:

fldDatasetID pk + fk

fldRegionID pk + fk

This table's values are essentially static.

PHEW!!

I have been through this structure several times and tried to re–jig things to avoid the circular relationships created, but none of the options that I have tried suits the requirements.

Where I have a problem is when I try and pre–populate items in the deployment table. I don't know how to set up my queries so that only valid deployment records may be added, given the values in both the Build table and the RegionDataset table.

EG:

RegionA receives dataset updates for Dataset1 and Dataset4

RegionB receives dataset updates for Dataset1, Dataset2 and Dataset3

This is represented in the RegionDataset table by the following:

RegionA–Dataset1

RegionA–Dataset4

RegionB–Dataset1

RegionB–Dataset2

RegionB–Dataset3

Dataset1 has a version update v111, so a new build record is created with a compound primary key:

Dataset1–v111

Once this entry is made, I'd like my database to be able to automatically add the only two valid entries to the Deployment table:

Circular relationships and limiting allowed values

RegionA-Dataset1-v111

RegionB-Dataset1-v111

But my queries must be incorrect, because this does not work.

Thanks to anyone who has read this far!

If anyone has done something similar before or can point me in the right direction it would be greatly appreciated. I have searched I-don t-know-how-many helpfiles, newsgroups and websites, but without any luck.

- ***Follow-Ups:***

- ◆ ***Re: Circular relationships and limiting allowed values***

- ◇ *From:* Tim Ferguson

- ◆ ***Re: Circular relationships and limiting allowed values***

- ◇ *From:* Vincent Johns

- ◆ ***Re: Circular relationships and limiting allowed values***

- ◇ *From:* Jeff Boyce

- Prev by Date: ***Re: AUTONUMBER- but with a custom alphanumeric***
- Next by Date: ***auto change spelling -how to disable***
- Previous by thread: ***Re: AUTONUMBER- but with a custom alphanumeric***
- Next by thread: ***Re: Circular relationships and limiting allowed values***
- Index(es):
 - ◆ ***Date***
 - ◆ ***Thread***